CMiner: Opinion Extraction and Summarization for Chinese Microblogs

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao

Abstract—Sentiment analysis of microblog texts has drawn lots of attention in both the academic and industrial fields. However, most of the current work only focuses on polarity classification. In this paper, we present an opinion mining system for Chinese microblogs called CMiner. Instead of polarity classification, CMiner focuses on more complicated opinion mining tasks - opinion target extraction and opinion summarization. Novel algorithms are developed for the two tasks and integrated into the end-to-end system. CMiner can help to effectively understand the users' opinion towards different opinion targets in a microblog topic. Specially, we develop an unsupervised label propagation algorithm for opinion target extraction. The opinion targets of all messages in a topic are collectively extracted based on the assumption that similar messages may focus on similar opinion targets. In addition, we build an aspect-based opinion summarization framework for microblog topics. After getting the opinion targets of all the microblog messages in a topic, we cluster the opinion targets into several groups and extract representative targets and summaries for each group. A co-ranking algorithm is proposed to rank both the opinion targets and microblog sentences simultaneously. Experimental results on a benchmark dataset show the effectiveness of our system and the algorithms.

Index Terms—Sentiment analysis, text mining, social network services

1 Introduction

Microblogging services such as Twitter, Sina Weibo and Tencent Weibo have swept across the globe in recent years. Users of microblogs range from celebrities to ordinary people. They express their emotions or attitudes towards a broad range of topics. It is reported that there are more than 340 million tweets per day on Twitter and more than 200 million messages on Sina Weibo. It is noteworthy that topics aggregated by the same hashtag play an important role in Chinese microblog websites. These websites often provide an individual webpage to list hot topics and invite people to participate in the discussion. The hot topics have a wide coverage of timely events and entities. Each of them may contain tens of thousands of messages. Mining the opinions in these topics will be valuable for monitoring the public opinion as well as helping microblog users to get a deeper overview of the topics.

Research of microblog sentiment analysis has been mainly conducted on polarity classification [1], [2], [3]. However, classifying microblog texts at the sentence level is often insufficient to understand the opinions in a topic. In this study, we propose a powerful opinion mining system for microblog topics – CMiner. CMiner mainly focuses on opinion target extraction and summarization for Chinese microblogs. Both of the two tasks have not been well investigated yet. Novel algorithms are developed and integrated into the end-to-end system. For each microblog topic, CMiner provides the users with different opinion target groups along with the opinion summary towards them. It also contributes as a pioneering exploration of applying the traditional aspect-based opinion summarization [4] to microblog topics.

Most of previous microblog sentiment analysis research focuses on Twitter and especially in English. However, the analysis of Chinese microblogs (i.e., Weibo) has some notable differences with that of Twitter: 1) Chinese word segmentation is a necessary step for analyzing Chinese texts, which is particularly difficult for microblogs. 2) The usage of hashtag is quite different for that in Twitter. Hashtags in English tweets are mostly used to highlight the sentiment information such as "#love", "#sucks" or serve as user-annotated coarse topics such as "#news", "#sports" [5]. However, in Chinese microblogs, most of the hashtags are only used to indicate fine-grained topics, such as #NBA总决赛第六场# (#NBAFinalG1#). 3) Hashtags in Twitter always appear within a sentence such as "I love #BarackObama!" while hashtags in Chinese microblogs are always isolated and are surrounded by two # symbols such as "#巴拉克奥巴马# 我爱他!" ("#BarackObama# I love him!").

CMiner employs a novel and efficient algorithm for opinion target extraction in microblog texts. In sentiment analysis, opinion target is the object to which the opinion is expressed. For example, in the sentence "The sound quality is good!", "sound quality" is the opinion target. Opinion target extraction has been mostly studied in customer review texts where opinion targets are often referred to as features or aspects [4]. Many opinion target extraction approaches rely on dependency parsing [6], [7], [8] and opinion target extraction is regarded as a domain-dependent task [9]. However, such approaches are not suitable for microblogs because existing natural language processing tools perform poorly on microblog texts due to their inherent characteristics. Studies show

• *The authors are with the Institute of Computer Science and Technology, the MOE Key Lab of Computational Linguistics, Peking University, Beijing, China. E-mail: {zhouxinjie, wanxiaojun, xiaojianguo}@pku.edu.cn*

Manuscript received 31 July 2015; revised 29 Feb. 2016; accepted 7 Mar. 2016. Date of publication 11 Mar. 2016; date of current version 1 June 2016. Recommended for acceptance by M. Sanderson. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2016.2541148

that one of the state-of-the-art part-of-speech taggers - OpenNLP only achieves the accuracy of 74 percent on tweets [10]. The syntactic analysis tool that generates dependency relation may perform even worse. Besides, microblog messages express opinion in different ways and they do not always contain opinion words, which lowers the performance of methods utilizing opinion words to find opinion targets.

In this study, we propose an unsupervised method to collectively extract the opinion targets from opinionated sentences in the same topic. We first present a dynamic programming based segmentation algorithm for Chinese hashtag segmentation. After that, all the noun phrases in each sentence and the hashtag segments are extracted as opinion target candidates. We assume that similar sentences in a topic may share the same opinion targets. Therefore, an unsupervised label propagation algorithm is proposed to collectively rank the candidates of all sentences. Finally, the candidates which get higher scores are selected as the opinion targets.

While finding the opinion targets in a topic is not enough, CMiner tries to generate opinion summaries for different opinion targets. We first categorize various opinion targets into several groups. Each opinion target group plays a similar role as a product aspect. For clustering the extracted opinion targets, we consider three different similarity measures – character similarity, context similarity and semantic similarity. Afterwards, we find the representative targets and generate summaries for each opinion target group. A co-ranking algorithm is proposed to rank the targets and the related microblog sentences simultaneously.

Our contributions in this study are summarized as follows:

1) We propose an end-to-end opinion mining system CMiner for Chinese microblogs. It integrates sentiment classification, opinion target extraction and opinion summarization techniques. We mainly focus on the latter two problems and propose novel algorithms to tackle the tasks.
2) We propose an unsupervised label propagation algorithm for collective opinion target extraction. It does not require any manually labeled data.
3) We propose the opinion target group based opinion summarization technique for microblogs, which is an extension of the traditional aspect-based opinion summarization method. For each opinion target group, a co-ranking algorithm is used to find the representative targets and sentences to form the opinion summary.
4) To the best of our knowledge, both opinion target extraction and opinion summarization have not been well studied in microblogs yet. It is more challenging than microblog sentiment classification as well as opinion mining in review texts.

2 Related Work

2.1 Opinion Target Extraction

Opinion target extraction is a fine-grained word-level task of sentiment analysis. When performed on customer review texts, opinion target extraction is also called aspect extraction. For example, "screen", "battery", and "color" are supposed to be extracted as aspects for "cell phone". The pioneering research on this task is conducted by [11], who proposed a method which extracted frequent nouns and noun phrases as the opinion targets via association mining. In review texts, opinion words and opinion targets always appear together. Their relation can be captured via dependency parsing. Zhuang et al. [6] identified such dependency patterns to discover valid aspect-opinion pairs. Qiu et al. [8] proposed a double propagation method to extract opinion word and opinion target in an iterative way.

In recent years, statistical topic models have emerged as a useful method for mining product aspects and opinion words. In these methods, aspects and opinion words are modeled as topics. Titov and McDonald [14] extended the standard topic modeling method LDA to induce multi-grain topics. They show that the global topics can discover entities while the local topics can discover aspects. Zhao et al. [15] improved the multi-grain topic model by modeling aspects and opinion words together. A max-entropy labeler is firstly trained to distinguish between aspects and opinion words.

In addition to customer review texts, opinion target extraction has been conducted on open-domain texts such as news articles. Kim and Hovy [12] used semantic role labeling as an intermediate step to label opinion holder and target. They utilized FrameNet data to get annotated corpus by mapping target words to opinion-bearing words and mapping semantic roles to holders and targets. Ma and Wan [16] extracted opinion targets from news comments using Centering Theory. Their approach uses global information in news articles as well as contextual information in adjacent sentences of comments. Yang and Cardie [17] jointly extracted the opinion expressions, the opinion holders, and the targets of the opinions, and the relations. Their approach is evaluated based on a standard corpus for fine-grained opinion analysis - the MPQA corpus [18] and the results outperform traditional baselines by a significant margin.

In this study, we address the opinion target extraction problem in microblogs. It is worth noting that the task is different from the work introduced above and faces new challenges. Compared to aspect extraction, our targeting data become open-domain microblog topics while aspect extraction focuses on a certain product (domain). Besides, aspect extraction aims to find a lexicon of aspects of a given product while we aim to find the opinion target of each microblog message. In addition, opinion targets (i.e., aspects) in product reviews are always noun words or short noun phrases. However, in microblogs, opinion targets can be long phrases formed by two or more words. The multi-word property of microblog opinion targets also restricts the usage of topic modeling methods. Besides, many aspect extraction methods rely on the dependency relation between opinion word and opinion target. Some of them directly use the dependency path as patterns. Such approach is also not applicable to microblogs due to the following two reasons: 1) Dependency parsing results on microblog texts are very noisy and unreliable. 2) Many opinionated sentences in Chinese microblogs do not contain any opinion word. Sequence labeling methods have also been applied to opinion target extraction [7], [17], which require annotated corpus to train a labeler. However, it is hard to get training data for microblogs. Fine-grained annotation for opinion targets will consume lots of human effort.

Besides, microblogs contain lots of new words and are evolving all the time. It is difficult to create a corpus which has broad coverage.

Faced with the above problems, we propose an unsupervised method to extract opinion targets in microblogs. We also develop an effective algorithm for hashtag segmentation and try to leverage an online encyclopedia to improve the Chinese word segmentation performance on microblogs.

2.2 Opinion Summarization

Opinion summarization is the study that attempts to generate a concise and digestible summary of a large number of opinions [19]. The most common technique for this task is aspect-based opinion summarization, which generates summaries for a set of opinion targets or sub-topics. For example, we can summarize the opinions towards a cell phone in terms of different aspects such as screen, battery life or signals. Hu and Liu [11] proposed the original framework for it. After finding the aspects of a product in the reviews, they gave the number and percent of people who hold positive and negative opinions about the entities and aspects. The resulting opinion summary is a form of structured summary that contains aspects, opinion distribution towards each aspect and individual review sentences. Since then, the aspect-based approaches become very popular and have been heavily explored over the last few years. Liu et al. [20] proposed the system Opinion Observer, which shows statistics of opinion orientation in each aspect and even enables users to compare opinion statistics of several products. Carenini et al. [21] presented and compared two approaches for the task of summarizing evaluative arguments. The first one is a sentence extraction-based approach while the second one is a language generation-based approach. Kim and Zhai [22] proposed the contrastive opinion summarization problem. They aim to find reviews that have opposite sentiment orientations on the same aspect. The task is formulated as an optimization problem and two general methods are proposed for generating a comparative summary using the content similarity and contrastive similarity of two sentences. Lu et al. [23] ordered aspects and their corresponding sentences based on a coherence measure, which tried to optimize the ordering so that they could best follow the sequences of aspect appearances in their original posting.

In addition to review summarization, opinion summarization has been applied on microblogs recently. Weng et al. [24] presents a system to summarize a microblog post and its responses with the goal to provide readers with a more constructive and concise set of information for efficient digestion. They proposed a novel two-phase summarization scheme. In the first phase, the post plus its responses are classified into one of the following four categories, interrogation, sharing, discussion and chat. Opinion analysis is then used to classify the polarity of the sharing and discussion posts. Bora [25] built a sentiment classification tool which is used to analyze a collection of tweets. They give the opinion distribution to of the tweets retrieved by a given query as the summary. Meng et al. [26] also dealt with the opinion summarization problem for a given entity. After getting the tweet collection which contains the entity, they extract the subtopics from all the hashtags in the tweets. Each subtopic can be represented by several hashtags. For each subtopic, a classifier is used to find insightful tweets which not only convey opinions but also provide insight. They also build a SVM-based classifier to find target-dependent opinions. It is worth noting that the task of [26] is different from ours. In their framework, the targeting corpus is the tweet collection retrieved by an entity. They extract different hashtags as subtopics from the tweet collection and summarize the opinion towards these topics. However, the input data of CMiner is a microblog topic denoted by a hashtag. We first identify the opinion targets in the topic and then summarize the opinion towards these targets. Meanwhile, Meng et al. [26] cannon not deal with Chinese microblogs because the hashtags related to an entity will be isolated and sparse. It is difficult to mine subtopics from the Chinese hashtags. Meanwhile, the patterns used to discover insightful tweets are not suitable for different languages.

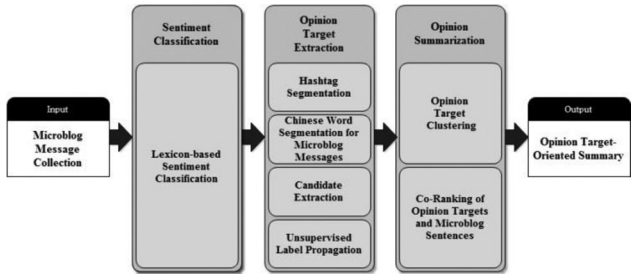


Fig. 1. The pipeline of CMiner.

Compared to existing work that summarizes opinion for an entity, CMiner is the first study that explores opinion summarization for microblog topics. It can provide users with an intuitive way to understand the opinions in a microblog topic. In addition, CMiner uses some specific methods to handle the unique characteristics of microblogs, such as opinion target clustering.

3 System Overview

The CMiner system follows the traditional pipeline of aspect-based opinion summarization as shown in Fig. 1. Firstly, we build a lexicon-based sentiment classifier. Since sentiment classification in microblogs has been heavily explored in recent years and is not the focus of this study, we simply adopt a lexicon-based strategy [27] which classifies a message into positive, negative or neutral using an opinion lexicon.

Our system mainly focuses on opinion target extraction and opinion summarization, both of which are more challenging than sentiment classification. For opinion target extraction, we firstly propose a dynamic programming based algorithm for Chinese hashtag segmentation. The hashtag segments are used to obtain named entities related to the topic from an online encyclopedia. These named entities and hashtag segments serve to improve the Chinese word segmentation results, which is crucial for the opinion target candidate extraction in the next step. Finally we rank the opinion target candidates via an unsupervised label propagation algorithm.

For opinion summarization, we adapt the traditional aspect based summarization framework to fit microblogs. Firstly, we cluster the extracted targets into semantically coherent groups because a microblog topic may contain tens or hundreds of different opinion targets. For each opinion target group, we need to find representative opinion

TABLE 1
Motivation Examples

Topic	Sentence
#官员财产公示# #Property publicity of government officials#	1. 纯属作秀! (Just for show!) 2. 财产公示在中国就是作秀。 (Property publicity is just a show in China.)
#菲军舰恶意撞击# #Philippine navy vessel hits Chinese fishing boat#	1. 政府还是不够强硬。 (The government is not tough enough.) 2. 政府为何不能强硬一些? (Why cannot the government take a tougher line?)

targets for the group and find representative microblog sentences as the summary. They are accomplished jointly via a co-ranking algorithm. Detailed strategies for opinion target extraction and summarization will be revealed in the following sections.

4 Microblog Opinion Target Extraction

4.1 Motivation

As described above, hashtags in Chinese microblogs often indicate fine-grained topics. In this study, we aim to collectively extract the opinion targets of messages with the same hashtag, i.e., in the same topic. The opinion target of a sentence can be divided into two types. One of which is *explicit target* that appears in the sentence such as "I love *Obama*". The other one is called *implicit target* that appears out of the sentence, for example, the sentence "Just for show!" in Table 1 comments on the target in the hashtag "#Property publicity of government officials#". Such implicit opinion targets are not considered in previous work and are more difficult to extract than explicit targets. However, we believe that the contextual information will help to locate both of the two kinds of opinion targets because similar sentences in a topic may share the same opinion target, which provides the possibility for collective extraction.

Table 1 shows the motivation examples of two topics and four sentences. The two sentences in each topic are considered to be similar because they share several Chinese words. In the topic #官员财产公示# (#Property publicity of government officials#), the first sentence omits the opinion target. However, the second one contains an explicit target "财产公示" ("property publicity") in the sentence. If we find the correct opinion target for sentence 2, we can infer that sentence 1 may have an implicit opinion target similar to the opinion target in sentence 2. In the second topic, both sentences contain a noun word "政府" ("government"). The similarity between these two sentences may indicate that both of the two sentences are expressing opinion on "政府" ("government").

Based on the above observation, we can assume that similar sentences in a topic may have the same opinion targets. Such assumption can help to locate both explicit and implicit opinion targets. Following this idea, we firstly extract all the noun phrases in each sentence as opinion target candidates. Afterwards, an unsupervised label propagation algorithm is proposed to rank these candidates for all sentences in the topic.

4.2 Context-Aware Hashtag Segmentation

In our approach, the Chinese word segmentations of hashtags and topic contents are treated separately. Existing Chinese word segmentation tools work poorly on microblog texts. The segmentation errors especially on opinion target words will directly influence the results of part-of-speech tagging and candidate extraction. However, some of the opinion target words in a topic are often included in the hashtag. By finding the correct segments of a hashtag and adding them to the user dictionary of the Chinese word segmentation tool, we can remarkably improve the overall segmentation performance.

The following example can help to understand the idea better. In the topic #90后打老人# (means "A young man hits an old man"), "90后" (literally "90 later" and means a young man born in the 90s) is an important word because it is the opinion target of many sentences. However, existing Chinese word segmentation tools will regard it as two separate words "90" and "后" ("later"). Afterwards, in the part-of-speech tagging stage, "90" will be tagged as a number and "后" ("later") will be tagged as a localizer. As we only extract noun phrases as opinion target candidates, the wrong segmentation on "90后" makes it impossible to find the right opinion target. Such error may occur many times in sentences that mention the word "90后" and express opinion on it. To solve the problem, CMiner firstly utilizes the topic context to identify the out-of-vocabulary words in the hashtag. For example, the high frequency of "90后" in the context is a strong indication that it should be regarded as a single word. After segmenting the hashtag correctly into "90后/打/老人", we can add the hashtag segments to the user dictionary of the segmentation tool to further segment the texts of the whole topic.

The basic idea for our hashtag segmentation algorithm is to regard strings that appear frequently in a topic as words. Formally, given a hashtag h that contains n Chinese characters $c_1c_2...c_n$. We want to segment into several words $w_1w_2...w_m$, where each word is formed by one or more characters.

Firstly, we define the stickiness score for a Chinese string $c_1c_2...c_n$ based on the Symmetrical Conditional Probability (*SCP*) [28]:

$$SCP(c_1c_2...c_n) = \frac{\Pr(c_1c_2...c_n)^2}{\frac{1}{n-1}\sum_{i=1}^{n}\Pr(c_1...c_i)\Pr(c_{i+1}...c_n)}, \quad (1)$$

and $SCP(c_1) = \Pr(c_1)^2$ for string with only one character. $\Pr(c_1c_2...c_n)$ is the occurrence frequency of the string in all the messages of the topic.

Following [29], we smooth the SCP value by taking logarithm calculation. Besides, the length of the string is taken into consideration,

$$SCP'(c_1c_2...c_n) = n \times \log SCP(c_1c_2...c_n), \quad (2)$$

where n is the number of characters in the string.

Then the stickiness score is defined by the sigmoid function as follows:

$$Stickiness(c_1c_2...c_n) = \frac{2}{1 + e^{-SCP'(c_1c_2...c_n)}}. \quad (3)$$

For the hashtag $h = c_1c_2...c_n$, we want to segment it into m words $w_1w_2...w_m$ which maximize the following equation,

$$\max \sum_{i=1}^{m} Stickness(w_i). \quad (4)$$

The optimization of (4) can be solved efficiently by dynamic programming which iteratively segments a string into two substrings. As shown in Algorithm 1, the output *seg* is a two-dimensional array where *seg*[*i*, *j*] shows the position to segment the string $c_ic_{i+1}...c_j$. For example, if *seg*[*i*, *j*] is equal to *k*, we should segment $c_ic_{i+1}...c_j$ into two substrings $c_ic_{i+1}...c_k$ and $c_{k+1}c_{k+2}...c_j$. Different from [29] which calculates the SCP value of each string based on Microsoft Web N-Gram, our hashtag segmentation algorithm only uses the topic context and does not need any additional corpus.

4.3 Finding New Words from Online Encyclopedia

Through hashtag segmentation, we find several important words and successfully identify new words in the hashtag. However, the microblogs messages may contain much more new words which are hard to segment for current Chinese word segmentation tools. For example, in the topic #曼联vs皇马# (#Manchester United versus Real Madrid#), our hashtag segmentation algorithm segments it into three words "曼联" ("Manchester United"), "vs", and "皇马" ("Real Madrid"). The microblog messages contain lots of names such as "C罗" ("C. Ronaldo"). An easy way to correctly segment these words is to use a dictionary to assist the Chinese word segmentation tool. In this study, we leverage the online encyclopedia Baidu Baike to get the named entities related to the topic.

Baidu Baike is currently one of the largest online Chinese encyclopedias. Up to now, it contains more than six million entries while Chinese Wikipedia has only seven hundred thousand entries[1]. Besides, articles on Baidu Baike update quickly. Events or people related to hot topics will be created in a short time.

In our approach, all the hashtag segments and the hashtag itself are used as queries to search for entries on Baidu Baike. For example, we use the following four queries "曼联" ("Manchester United"), "vs", "皇马" ("Real Madrid") and "曼联vs皇马" ("Manchester United versus Real Madrid") for the topic #曼联vs皇马#. For each query, we download the articles of the top three retrieved entries. The named entities in each article which have a hyperlink to another entry are recorded. These named entities are called *First Layer NE*. For all the *First Layer NE*, we download their articles on Baidu Baike and extract the *Second Layer NE*.

The *First Layer NE* and *Second Layer NE* together with the hashtag segments are added into the user dictionary of the Chinese word segmentation tool ICTCLAS. ICTCLAS is used to segment all the microblog messages and get the part-of-speech tags.

1. http://zh.wikipedia.org/wiki/Special:Statistics

4.4 Candidate Extraction

In our approach, noun phrases are treated as opinion target candidates. We extract the noun phrases in each sentence by the following regular expression: *(noun|adj)(noun|adj|的)*noun*. That means a noun phrase can only include nouns, adjectives and the Chinese word "的" ("of"). It should begin with a noun or adjective and end with a noun. For example, in the following sentence, "中国/n 的/u 教育/n 制度/n 有/v 问题/n 。/w" ("Chinese education system has problems."), "中国的教育制度" ("Chinese education system") and "问题" ("problem") are extracted as noun phrases.

Algorithm 1 Dynamic Programming Algorithm for Hashtag Segmentation

Input
hashtag: $h = c_1c_2...c_n$
Output
segmentation position array: *seg*
//assign the *Stickiness* value to score
for $i = 1$ to n **do**
  for $j =$ i to n **do**
    seg[*i*, *j*] = *j*
    score[*i*, *j*] = *Stickiness*($c_1c_2...c_n$)
  end for
end for
//find the segmentation position **for** $c_ic_{i+1}...c_j$ which maximize score[*i*, *j*]
for $i = 1$ to n **do**
  for $j = i$ to n **do**
    for $m = i$ to j **do**
      if *score*[*i*, *m*] + *score*[*m*+1, *j*] > *score*[*i*, *j*]
        score[*i*, *j*] = *score*[*i*, *m*] + *score*[*m*+1, *j*]
        seg[*i*, *j*] = *m*
      end if
    end for
  end for
end for

The length of a noun phrase is limited between two and seven Chinese characters. For each sentence, all phrases that match the regular expression and meet the length restriction are extracted as explicit opinion target candidates. The hashtag is regarded as an implicit candidate for all sentences. Besides, some opinionated sentences in microblogs do not contain any noun phrase, such as "无聊至极" ("*So boring*!"). These sentences may express opinions on an object that has been mentioned before. Therefore, the explicit candidates of the previous sentence in the same message are also taken as the implicit candidates for such sentences.

We do not use any syntactic parsing tool to extract noun phrases because the parsing results on microblogs are not reliable. A performance comparison of our rule based method and the state-of-the-art syntactic parser will be shown in Section 7.

4.5 Unsupervised Label Propagation for Candidate Ranking

We simply assume that each opinionated sentence has one opinion target, which is consistent with the statistical result of our dataset that over 93 percent sentences have only one

opinion target and each sentence has an average of 1.09 targets. Therefore, the most confident candidate of each sentence will be selected as the opinion target. In this section, we introduce an unsupervised graph-based label propagation algorithm to collectively rank the candidates of all sentences in a topic.

Label propagation [30], [31] is a semi-supervised algorithm which spreads label distributions from a small set of nodes seeded with some initial label information throughout the graph. The basic idea is to use information from the labeled nodes to label the adjacent nodes in the graph. However, our idea is to use the connection between different nodes to find the correct labels for all of them. Our unsupervised label propagation algorithm is summarized in Algorithm 2. Sentences are regarded as nodes and candidates of each sentence are regarded as labels. The label vector for each node is initialized based on the results of the candidate extraction step, which means no manually-labeled instances are needed in our model. In each iteration, the label vector of one node is propagated to the adjacent nodes. Both the sentence (node) similarity and the candidate (label) similarity are considered during propagation. Finally, we select the candidate with the highest score in the label vector as the opinion target for each sentence. The detail of the algorithm is presented in Algorithm 2.

Formally, an undirected graph $G = \langle V, E, \tilde{W} \rangle$ is built for each topic. A node $v \in V$ represents a sentence in the topic and an edge $e = (a, b) \in E$ indicates that the labels of the two vertices should be similar. $\tilde{W}$ is the normalized weight matrix to reflect the strength of this similarity. The similarity between two nodes W_{ab} is simply calculated by using the cosine measure [32] of the two sentences

$$W_{ab} = cos(T_a, T_b) = \frac{T_a \cdot T_b}{\|T_a\| \cdot \|T_b\|}. \tag{5}$$

where T_a and T_b are the term vectors of sentences a and b represented by the standard vector space model and weighted by term frequency. After calculating the similarity matrix W, we get the weight matrix $\tilde{W}$ by normalizing each row of W such that $\sum_b \tilde{W}_{ab} = 1$.

Algorithm 2. Unsupervised Label Propagation

Input:
Graph: $G = \langle V, E, \tilde{W} \rangle$
Candidate Similarity: $S \in R_+^{M \times M}$
Prior Labeling: $Y_v \in R_+^{1 \times M}$ for $v \in V$
Filtering Matrix: $F_v \in R_+^{M \times M}$ for $v \in V$
Probability: p^{inj} and p^{cont}
Output:
Label Vector: $\hat{Y}_v \in R_+^{1 \times M}$

for all $v \in V$ **do**
  $\hat{Y}_v \leftarrow Y_v$
end for
repeat
  for all $v \in V$ **do**
    $D_v \leftarrow \sum_{u \in V, u \neq v} \tilde{W}_{uv} (\hat{Y}_u \times S) \times F_v$
    $\hat{Y}_v \leftarrow p^{inj} Y_v + p^{cont} D_v$
  end for
until convergence

$$Y_v = \begin{bmatrix} 1 & 1 & 0.5 & 0 \end{bmatrix} \rightarrow F_v = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 2. An example of the filtering matrix.

For each sentence (node) v, a candidate set C_v is extracted in the previous step. The candidate set CT for the whole topic is the union of all C_v

$$CT = \bigcup C_v. \tag{6}$$

The total number of candidates in the topic is denoted by $M = |CT|$. We calculate the candidate similarity matrix $S \in R_+^{M \times M}$ based on Jaccard Index:

$$S_{ij} = \frac{|A(CT_i) \cap A(CT_j)|}{|A(CT_i) \cup A(CT_j)|} \qquad 1 \leq i \neq j \leq M, \tag{7}$$

where $A(CT_i)$ and $A(CT_j)$ are the Chinese character sets of the ith and jth candidates in CT respectively

Candidates are regarded as labels in our model and without loss of generality we assume that the possible labels for the whole topic are $L = \{1 \ldots M\}$ and each label in L corresponds to a unique candidate in CT. For each node $v \in V$, a label vector $Y_v \in R_+^{1 \times M}$ is initialized as

$$(Y_v)_k = \begin{cases} w & L_k \in C_v \\ 0 & L_k \notin C_v \end{cases} \qquad 1 \leq k \leq M, \tag{8}$$

where w is the initial weight of the candidate. We set $w = w_e$ if L_k is an explicit candidate (extracted noun phrase) of v and $w = w_i$ if L_k is an implicit candidate (hashtag segment or inherited from previous sentence) of v. If L_k is not a candidate of the current sentence, then the corresponding value in the label vector is 0. These values which are initialized as zero should always remain zero during the propagation algorithm because the corresponding label does not belong to the candidate set C_v of node v. To reset the values on these positions, a diagonal matrix $F_v \in R_+^{M \times M}$ is created for all nodes v

$$(F_v)_{kk} = \begin{cases} 1 & (Y_v)_k > 0 \\ 0 & (Y_v)_k = 0 \end{cases} \qquad 1 \leq k \leq M, \tag{9}$$

where the subscript kk denotes the kth position in the diagonal of matrix F_v. We can right-multiply Y_v by F_v to clear the values of the invalid candidates. Fig. 2 shows an example of creating the filtering matrix for a label vector.

The propagation process is formalized via two possible actions: inject and continue, with pre-defined probabilities p^{inj} and p^{cont}. Their sum is unit: $p^{inj} + p^{cont} = 1$. In each iteration, every node is influenced by its adjacent nodes. The propagation influence for each node v is

$$D_v = \sum_{u \in V, u \neq v} \tilde{W}_{uv} (\hat{Y}_u \times S) \times F_v, \tag{10}$$

where $\hat{Y}_u$ is the label vector of node u at the previous iteration. By multiplying the candidate similarity matrix S, we aim to propagate the score of the ith candidate of node u not

only to the ith candidate of node v, but also to all the other candidates. W_{uv} measures the strength of such propagation. The filtering matrix F_v is used to clear the values of the invalid candidates as described above.

Then the label vector of node v is updated as follow,

$$\hat{Y}_v = p^{inj} Y_v + p^{cont} D_v. \tag{11}$$

When the positions of the largest values in all label vectors keep unchanged in ten iterations, it is regarded that the algorithm has already converged.

5 Microblog Opinion Summarization

5.1 Opinion Target Clustering

In a microblog topic, users express opinions on lots of different targets related to the topic. When the number of microblog messages grows to thousands or hundreds of thousands, the number of opinion targets will also be unacceptably large to display to the users. Our CMiner system tries to cluster these opinion targets into semantically coherent groups and selects the important opinion target groups to display.

After extracting the opinion targets of all the microblog sentences in the previous stage, CMiner uses three different similarity measures to get the relatedness between opinion targets, namely character similarity, context similarity and semantic similarity.

The character similarity considers how many Chinese characters appear in both of two opinion targets. For example, the opinion targets "我们的政府" (our government), "这个政府" (this government) and "政府" (government) refer to the same entity. Such kind of relatedness can be captured via character similarity because all the opinion targets contain the same characters "政府". Formally, given two opinion targets OT_1 and OT_2, we get the character similarity based on Jaccard Index:

$$CharSim(OT_1, OT_2) = \frac{|A(OT_1) \cap A(OT_2)|}{|A(OT_1) \cup A(OT_2)|}, \tag{12}$$

where $A(\cdot)$ is the Chinese character set of an opinion target. Note that the method used here is the same as that used for calculating the candidate similarity.

The second relatedness measure is context similarity. We think that the similarity of two opinion targets can be decided by their contexts. For each opinion target, we find all the sentences that contain the target and concatenate them as a context document. The document's tf-idf vector is regarded as the context vector for the opinion target. Therefore, for the two targets OT_1 and OT_2, whose context vectors are denoted as $\vec{C}_1$ and $\vec{C}_2$, cosine similarity is used to get the context similarity for two opinion targets,

$$CtxSim(OT_1, OT_2) = \frac{\vec{C}_1 \cdot \vec{C}_2}{|\vec{C}_1| \cdot |\vec{C}_2|}. \tag{13}$$

Lastly, we consider the semantic relatedness via explicit semantic analysis (ESA) [33]. Each opinion target is mapped to a high-dimensional vector which is spanned by a Wikipedia database. We use $\vec{E}_1$ and $\vec{E}_2$ to represent the ESA vectors of two opinion targets, and then the semantic similarity is defined as

$$SemSim(OT_1, OT_2) = \frac{\vec{E}_1 \cdot \vec{E}_2}{\left|\vec{E}_1\right| \cdot \left|\vec{E}_2\right|}. \tag{14}$$

To aggregate the above three similarity measures, we simply take the arithmetic average of them,

$$Sim(OT_1, OT_2) = \frac{CharSim(OT_1, OT_2) + CtxSim(OT_1, OT_2) + SemSim(OT_1, OT_2)}{3}. \tag{15}$$

Algorithm 3. Co-ranking of Opinion Tragets and Sentences

Input
Transition Matrices: $\tilde{M}, \tilde{N}, ST$ and TS
Convergence Threshold ε
Parameters α and λ
Output
Score Vectors: v_s and v_t

$v_t = \frac{1}{|T|} \cdot \vec{1}$
$v_s = \frac{1}{|S|} \cdot \vec{1}$
Repeat
  $v_t^{z+1} = \lambda \cdot \tilde{M} \cdot v_t^z + (1-\lambda) \cdot ST \cdot v_s^z$
  $v_s^{z+1} = \lambda \cdot \tilde{N} \cdot v_s^z + (1-\lambda) \cdot TS \cdot v_t^z$
Until $|v_t^{z+1} - v_t^z| < \varepsilon$ and $|v_s^{z+1} - v_s^z| < \varepsilon$
$v_t = v_t^{z+1}$
$v_s = v_s^{z+1}$

Afterwards, we adopt the state-of-the-art clustering algorithm Affinity Propagation [34] to cluster all the extracted opinion targets. It does not require a pre-determined cluster number which is important for our task, because the cluster number of opinion targets may vary significantly from topic to topic.

5.2 Co-Ranking of Opinion Targets and Microblog Sentences

CMiner aims to summarize the opinions for each opinion target group by selecting representative sentences. Meanwhile, we need to display each opinion target group with several representative opinion targets. We propose a graph-based co-ranking algorithm following the framework of [35] to select representative summary sentences and opinion targets simultaneously. The main intuition behind co-ranking is that there is a mutually reinforcing relationship between opinion targets and microblog sentences. For the homogeneous relationship, an opinion target (or a sentence) can be representative if it is connected to other representative targets (or sentences). For the heterogeneous relationship, important opinion targets should appear together with important sentences. Following the PageRank paradigm, the homogeneous relationship can be formulated as two different random walks, one of which among the opinion targets and the other among the microblog sentences. The heterogeneous relationship can be formulated as a random walk between

opinion targets and microblog sentences. CMiner integrates both the homogeneous relationship and heterogeneous relationship into a joint ranking framework.

Specially, for each opinion target group T which is obtained in the above section, we define the transition matrix $M \in R^{|T|\times|T|}$ for the opinion target graph as follows

$$M_{ij} = \frac{Sim(T_i, T_j)}{\sum_{k=1}^{|G|} Sim(T_i, T_k)}, \tag{16}$$

where T_i is the ith opinion target in group T, $|T|$ is the total number of opinion targets in T, the *Sim* function is defined in (7). Consider a random walk on the opinion target graph, at each step we do not make a usual random walk step, but instead jump to any vertex with probability α, chosen uniformly at random. The new transition matrix becomes

$$\tilde{M} = (1-a)M + \frac{1}{|T|}\vec{1}\cdot\vec{1}\,T, \tag{17}$$

where $\vec{1} \in R^{1\times|G|}$ is the row vector with all the values equal to 1.

For each opinion target group T, we select all the microblog sentences which contain at least one opinion target in T and denote the sentence set as S. We can derive the similar transition matrix for the sentence graph as

$$\tilde{N} = (1-a)N + \frac{1}{|S|}\vec{1}\cdot\vec{1}\,T, \tag{18}$$

where $|S|$ is the total number of sentences, the matrix $N \in R^{|S|\times|S|}$in the above equation is defined as

$$N_{ij} = \frac{CosSim(S_i, S_j)}{\sum_{k=1}^{|S|} CosSim(S_i, S_k)}. \tag{19}$$

The bipartite relationship between the opinion target graph and the microblog sentence graph is build based on the occurrence of an opinion target in a sentence. Namely, the values in its adjacency matrix $E \in R^{|T|\times|S|}$ are the values of the indicator function of an opinion target being included in a sentence, i.e.,

$$E_{ij} = \mathbb{I}(T_i \text{ is in } S_j). \tag{20}$$

Based on E, we can define the transition matrix from the opinion target graph to the microblog sentence TS and the transition matrix from the microblog sentence to the opinion target graph ST,

$$TS_{ij} = \frac{E_{ij}}{\sum_{k=1}^{|T|} E_{ik}}, \tag{21}$$

$$ST_{ij} = \frac{E_{ji}}{\sum_{k=1}^{|S|} E_{ki}}. \tag{22}$$

Our co-ranking scheme is summarized in Algorithm 3, which combines the random walk in the opinion target graph and in the sentence graph together. For a random surfer (RS) on the graph, it can choose to travel though a homogeneous edge (from target to target or from sentence to sentence) with probability $\boldsymbol{\lambda}$ and choose to travel though a heterogeneous edge (from target to sentence or from sentence to target) with probability 1-$\boldsymbol{\lambda}$. Initially, all the opinion targets are assigned the weight $1/|\mathrm{T}|$. Similarly, the initial score vector for microblog sentences is $1/|\mathrm{S}|$. During the random walk process, v_{t} and v_{s} are updated as follows,

$$v_t^{z+1} = \lambda \cdot \tilde{M} \cdot v_t^z + (1-\lambda)\cdot ST \cdot v_s^z, \tag{23}$$

$$v_s^{z+1} = \lambda \cdot \tilde{N} \cdot v_s^z + (1-\lambda)\cdot TS \cdot v_t^z. \tag{24}$$

After the process converges, we select opinion targets with top ranking values in the vector v_{t} as representatives for the group, and select microblog sentences with top ranking values in the vector v_{s} as the opinion summary. The readers can refer to [35] for the convergence and complexity analysis of the algorithm.

TABLE 2
Detailed Statistics of the CMSAE Dataset

# of microblog messages	17,518
# of microblog sentences	31,675
# of annotated messages	2,000
# of annotated sentences	3,416
# of opinionated sentences	2,152
# of opinion targets	2,361

6 Experiments on Opinion Target Extraction

6.1 Dataset

We use the benchmarking dataset of the 2012 Chinese Microblog Sentiment Analysis Evaluation (CMSAE)[2] held by the China Computer Federation (CCF). The evaluation contains the task of opinion target extraction for Chinese microblogs.

The dataset contains 20 topics collected from Tencent Weibo, a popular Chinese microblogging website. The 20 topics in the dataset cover many of the most sensitive and pressing issues in China, such as government corruption, food safety, education, moral degradation, and so on. Detailed statistics of the dataset are listed in Table 2.

6.2 Evaluation Metric

Precision, recall and F-measure are used in the evaluation. Since expression boundaries are hard to define exactly in annotation guidelines [18], both the strict evaluation metric and the soft evaluation metric are used in CMSAE.

Strict evaluation: For a proposed opinion target, it is regarded as correct only if it covers the same span with the annotation result. Note that, in CMSAE, an opinion target should be proposed along with its polarity. The correctness of the polarity is also necessary.

Soft evaluation: The soft evaluation metric presented in [37] is adopted by CMSAE. The *span coverage c* between each pair of the proposed target span s and the gold standard span s' is calculated as follows,

$$c(s, s') = \frac{|s \cap s'|}{|s'|}. \tag{25}$$

2. http://tcci.ccf.org.cn/conference/2012/pages/page04_eva.html. The dataset can be publicly accessed on the website.

TABLE 3
Comparison Results with Baseline Methods

Method	Strict			Soft		
	P	R	F	P	R	F
AssocMi	0.22	0.20	0.21	0.47	0.43	0.45
CRF-C	0.59	0.15	0.24	0.70	0.18	0.28
CRF-S	0.61	0.27	0.35	0.73	0.31	0.41
Ours	0.43	0.39	0.41	0.61	0.55	0.58

In (25), the operator $|\cdot|$ counts Chinese characters, and the intersection $\cap$ gives the set of characters that two spans have in common.

Using the span coverage, the span set coverage C of a set of spans S with respect to another set S' is

$$C(S, S') = \sum_{s \in S} \sum_{s' \in S'} c(s, s'). \tag{26}$$

The soft precision P and recall R of a proposed set of spans $\hat{S}$ with respect to a gold standard set S is defined as follows:

$$\text{Precision} = \frac{C(\hat{S}, S)}{\left\|\hat{S}\right\|} \quad \text{Recall} = \frac{C(\hat{S}, S)}{\|S\|}. \tag{27}$$

The operator $\|\cdot\|$ in (27) counts the number of spans. The soft F-measure is the harmonic mean of soft precision and recall.

6.3 Comparison Results with Baseline Methods

We firstly introduce the following baseline models used for comparison.

AssocMi: We implement the unsupervised method for opinion target extraction based on [11]. Firstly, it regards the nouns or noun phrases which appear frequently in reviews as frequent opinion targets. Afterwards, the nearest noun or noun phrase of an opinion word in the sentence is also extracted as infrequent targets.

CRF: The CRF-based method used in [7] is also used for comparison. We implement both the single-domain and cross-domain models. Both models are evaluated using 5-fold cross-validation. More specifically, the single-domain model, denoted as *CRF-S*, trains different models for different topics. In each cross-validation round, 80 percent of each topic is used for training and the other 20 percent is used for test. The cross-domain model, denoted as *CRF-C*, uses 16 topics for training and the rest 4 topics for test in each round. Note that the single domain model *CRF-S* is not practical for real application because we cannot get training data for all the topics on microblogging websites. The following features are used in CRF-S and CRF-C, token, part-of-speech tag, shortest dependency path to the opinion expresssion in the sentence, word distance (whether the token is in the closest noun phrase regarding word distance to each opinion expression in a sentence) and opinion sentence (whether the sentence that the token belongs to contains an opinion expression). Since we do not have gold-standard opinion expression as [7], we use an opinion lexicon to identify opinion expressions in the sentences.

TABLE 4
Comparison Results with CMSAE Teams
(with Subjectivity and Polarity Classification)

Method.	Strict			Soft		
	P	P	F	P	R	F
Team-Avg	0.17	0.09	0.12	0.29	0.15	0.20
Team-3	0.26	0.16	0.20	0.40	0.25	0.31
Team-2	0.31	0.18	0.23	0.40	0.22	0.29
Team-1	0.30	0.27	0.29	0.39	0.36	0.37
Ours	0.37	0.27	0.32	0.48	0.37	0.42

Table 3 shows the experimental results of the four methods on CMSAE dataset. Our approach achieves the best result among them. AssocMi performs worst in strict evaluation but gets better results than the two CRF-based models in soft evaluation. The two CRF-based models achieve high precision but low recall. We can also observe that CRF-S is much more effective than CRF-C. It achieves better performance because it has already seen most of the opinion targets in the training set. However, it is impossible to build such single-domain model in practical applications because labeled instances are not available for new topics. Our proposed method does not require any training data and gets an increase of 17 percent over CRF-S and 70 percent over CRF-C in strict F-measure. In terms of soft F-measure, we achieve an increase of 41 and 107 percent over the two CRF models.

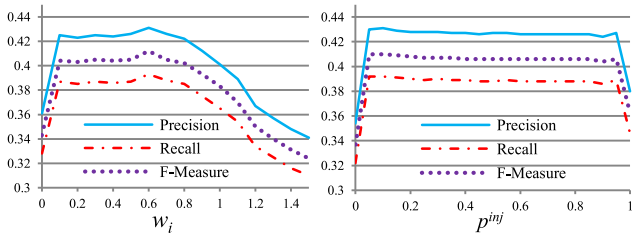
6.4 Comparison Results with Participating Teams of the Evaluations

We also compare our results with the teams which participated in the Chinese Microblog Sentiment Analysis Evaluation.

Sixteen teams participated in the opinion target extraction task of CMSAE. The methods of the top three teams are used as baselines here. They are denoted as Team-1, Team-2 and Team-3 respectively. The average result of all the sixteen teams is also included and is denoted as Team-Avg. We first briefly introduce the best team's method. The most important component of their model is a manually-built topic-dependent opinion target lexicon which is called object sheet. If a word or phrase in the object sheet appears in a sentence or a hashtag, it is extracted as the opinion target. The object sheet is manually built for each topic, which means their method cannot be applied to new topics and requires lots of human labor. Team-2 and Team-3 use similar methods which build several opinion target patterns based on the part-of-speech tags and dependency relations. However, their syntactic-based methods get much lower results than Team-1.

CMSAE requires all the teams to perform the subjectivity and polarity classification task in advance. The opinion targets are extracted only for opinionated sentences and should be proposed along with their polarity. To make a fair comparison, we directly use the subjectivity and polarity classification results of Team-1. Then our unsupervised label propagation method is used to extract the opinion targets for the proposed opinionated sentences. The parameters of our method are simply set as $p^{inj} = p^{cont} = 0.5$, $w_e = 1$ and $w_i = 0.5$.

Table 4 lists the comparison results with CMSAE teams. The average F-measure of all teams is 0.12 and 0.20 in strict



(a) Initial Candidate Weight (b) Injection Probability

Fig. 3. Influence of the parameters.

TABLE 5
Performance of Candidate Extraction and Opinion Target Extraction

Method	Total Candidates	Correct Candidates	F-Measure of OTE	
			Strict	Soft
Berkeley Parser	4554	877	0.36	0.56
Rule	4105	918	0.37	0.56
HS + Rule	4094	1042	0.41	0.57
BB+HS+Rule	4382	1053	0.41	0.58

and soft evaluation respectively. It shows that opinion target extraction is a quite hard problem in Chinese microblogs. Our method performs better than all the teams. It increases by 10 and 13 percent in the two kinds of F-measure compared to the best team.

6.5 Parameter Sensitivity Study

In this section, we study the parameter sensitivity. There are two major parameters in our algorithm: the initial weight w for both explicit and implicit candidates in (8) and the injection probability p^{inj} in (11).

The initial weights of explicit and implicit candidates are set differently because the explicit candidates are more likely to be the opinion targets. These two kinds of initial weights are denoted as w_e and w_i for explicit and implicit candidate, respectively. To study the impact of the initial weights, we fix w_e at 1 and tune w_i because we only care about the relative contribution of them. The injection probability is fixed at 0.5. Fig. 3a displays the opinion target extraction performance when w_i varies from 0 to 1.5.

In particular, when w_i is equal to 0, only explicit candidates are considered. When w_i becomes larger than 1, the implicit candidates become more important than explicit candidates. From the curve in Fig. 3a, we can observe that the implicit candidates help to improve the performance on the dataset significantly when w_i varies from 0 to 0.1. The performance reaches the peak when $w_i = 0.7$ and declines rapidly when w_i gets larger than 1.

To study the impact of injection probability p^{inj}, we fix the initial weights for explicit and implicit candidates as 1 and 0.5, respectively. Fig. 3b shows the results of opinion target extraction with respect to different values of the injection probability. We can observe that the performance keeps steady except for the two extreme values 0 and 1. The performance declines fast when p^{inj} gets close to 1, because the neighborhood information is not exploited in this setting. From the above two figures, we can conclude that our proposed method performs well and robustly with a wide range of parameter values.

6.6 Analysis of Candidate Extraction

Candidate extraction is an important step in our proposed method. If the correct opinion target is not extracted as a candidate, the ranking step will be in vain. As described in Section 3, we develop a hashtag segmentation algorithm and use a rule based method to extract noun phrases from each sentence. Furthermore, the named entities on Baidu Baike are exploited to improve the Chinese word segmentation performance. We do not use any parsing tool because we believe that the performance of these tools is not good enough when applied on microblogs. A quantitative comparison is shown in this section. Since we do not have gold-standard segmented words or noun phrases, we evaluate the performance based on the results of opinion target extraction.

We use one of the state-of-the-art syntactic analysis tools - *Berkeley Parser* [40] for comparison here. Noun phrases are directly extracted from the parsing results. *Rule* directly uses ICTCLAS to segment the whole topic content and labels each word with its part-of-speech tag. The method *HS+Rule* leverages the *hashtag segments* to enhance the segmentation result and also extracts explicit candidate using *Rule*. *BB+HS+Rule* leverages the named entities on *Baidu Baike* to enhance Chinese word segmentation.

Performance on candidate extraction is compared in Table 5. The second column shows the number of all extracted candidates for all the opinionated sentences by different methods. The third column shows the number of correct opinion targets among them. We can find that all of our rule-based methods outperform Berkeley Parser. The hashtag segments bring considerable improvement. The *HS+Rule* and *BB+HS+Rule* methods can find 14 percent more correct opinion targets than *Rule*. It also proves the effectiveness of our hashtag segmentation algorithm. *HS+Rule* and *BB+HS+Rule* lead to similar performance on opinion target extraction. *BB+HS+Rule* extracts more correct candidates but also produce much noise. Meanwhile, the CMSAE dataset does not involve many named entities which also restrict the performance of *BB+HS+Rule*.

6.7 Error Analysis

The performance of our opinion target extraction algorithm is still far from perfect. The F-measure is lower than 0.6 even using the soft evaluation metric as shown in Table 3. Fisrt of all, the low performance is partially caused by the ambiguous boundary of an opinion target which is hard to decide even for the dataset labelers. For example, in the sentence "Our government still has a lot of room to improve", the right opinion target can be either "government" or "our government".

In terms of our algorithm, the error is introduced mainly in the two steps, candidate extraction and candidate ranking. In the candidate extraction step, the explicit opinion targets sometimes are not treated as candidates due to errors of Chinese word segmentation, part-of-speech tagging and the noun phrase identification pattern. For example, in the sentence "90后怎么，你们70, 60, 50就好了吗?" ("You people born in the 50s, 60s, and 70s are no better than those born in

the 90s"), the correct opinion target should be "你们70, 60, 50" ("you people born in the 50s, 60s, and 70s") which will not be extracted as a candidate because it contains only an pronoun and three numbers. It will not be regarded as a noun phrase by our pattern or any other parsing tool. For the implicit candiates, our strategy considers nouns or noun phrases only in the nearest previous sentence and only when there does not exist any explicit candidate. However, sentences which contain nouns or noun phrases may also discuss about an implicit opinion target and the implict target is not always in the nearest previous sentence.

Intuitively, the above problem can be solved by considering more explicit and implicit candidates. However, it will introduce much noise for the next step of candidate ranking. The performance of the label propagation algorithm will decline when the number of candidates grows larger. Through our experiments, we find that the current stragety is an optimal trade-off between the two steps.

7 Experiments on Opinion Summarization

7.1 Dataset

We still use the CMSAE dataset to evaluate the opinion summarization performance. For opinion target clustering, we manually labeled opinion target clusters for the dataset. The extracted opinion targets are manually labeled into several groups. Each topic gets 10.4 opinion target groups on average. For opinion summarization, we ask three labelers to manually score the generated summaries from our system the baseline methods. The detailed description is in Section 7.3.

7.2 Evaluation of Opinion Target Clustering

In our system, three different similarity measures are proposed to compute the relatedness between opinion targets. Afterwards, Affinity Propagation[3] is used to cluster them. To verify the effectiveness of our proposed similarity measure, we run the algorithm based on the three kinds of similarity measures separately. Specially, we denote the clustering algorithm based on character similarity as *AP-Char*, the clustering algorithm based on context similarity as *AP-Char*, the clustering algorithm based on semantic similarity as *AP-Sem* and our algorithm as *AP-All*.

We use precision, recall, F-Measure and Rand Index as evaluation metrics. Suppose there are a total of N opinion targets in our annotated dataset, we can get $N(N\text{-}1)/2$ opinion target pairs. In the following, we refer to an opinion target pair (OT_1, OT_2) as *connected pair* if OT_1 and OT_2 belong to the same cluster, and *disconnected pair* otherwise. We define *TP*, *TN*, *FP* and *FP* as follows, *TP*: number of true positive decisions which assign the two opinion targets in a *connected pair* with the same cluster label. *TN*: number of true negative decisions which assign the two opinion targets in a *disconnected pair* with different cluster labels. *FP*: number of false positive decisions which assign the two opinion targets in a *disconnected pair* with the same cluster label. *FN*: number of false negative decisions which assign the two opinion targets in a *connected pair* with different cluster labels.

3. Code available at http://genes.toronto.edu/index.php?q=affinity%20propagation.

TABLE 6
Performance of Opinion Target Clustering

Method	Rand Index	Precision	Recall	F-Measure
AP-Char	0.77	0.55	0.26	0.33
AP-Ctx	0.43	0.26	0.68	0.35
AP-Sem	0.77	0.54	0.34	0.40
AP-All	0.80	0.61	0.45	0.51

The Rand Index measures the percentage of decisions that are correct, which is simply the accuracy

$$RI = \frac{TP + TN}{TP + TN + FP + FN}. \tag{28}$$

Compared to Rand Index, precision, recall and F-Measure put more emphasize on true positive decisions,

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F = \frac{P + R}{2 \cdot P \cdot R}. \tag{29}$$

Table 6 shows the results for opinion target clustering. AP-All achieves the best results in terms of both F-Measure and Rand Index. It outperforms the baseline methods by a large margin on F-Measure. AP-Char gets high precision but low recall. The most serious problem of it is that the character similarity cannot recognize the connection between opinion targets which do not share any same character. However, the high precision reveals that the shared characters are an important feature to measure the similarity. In contrast, AP-Ctx shows high recall but low precision because the context-based method regards many irrelevant opinion targets to be similar. AP-Sem relies on the Wikipedia database. Some opinion targets in microblogs do not appear in any Wikipedia entry. These opinion targets get zero similarity to all the other opinion targets, which lead to the low recall of AP-Sem.

7.3 Evaluation of Opinion Summarization

After clustering the opinion targets into several groups, our co-ranking algorithm generates representative targets and summaries for each group. We set the parameter $\boldsymbol{\alpha}$ as 0.1 and $\boldsymbol{\lambda}$ as 0.2 following [35].

To evaluate the performance of our system, we compared our method with two state-of-the-art summartization algorithms *LexRank* [36] and *ILP* [37]. *LexRank* uses the graph-based algorithm to rank the sentences while *ILP* formulate the sentence extraction problem using integer linear programming. For each opinion target group, we first select the microblog sentences which contain at least one of the targets in the group. The two baseline methods are used to generate the summary for each target group based on these sentences. Therefore, each opinion target group gets three different summaries, one from our method, one from ILP and the other one from LexRank.

We asked three labelers (who are not among the au-thors of this paper) to manually score these summaries according to four apsects, readability *(R)*, content responsiveness (*CR*), target-relatedness (*TR*) and overall responsiveness (*OR*). The three aspects readability, content responsiveness and overall responsiveness are adopted from DUC 2006 [38]. We add the target-relatedness aspect to measure whether

TABLE 7
Performance of Opinion Summarization

Method	R	TR	CR	OR
LexRank	2.10	2.04	1.59	1.76
ILP	2.93	3.46	2.87	2.94
Ours	4.09	3.88	3.71	3.76

the summary is closely related the current opinion target group. Each aspect is rated with scores from 1 (poor) to 5 (good). During the labeling procedure, each labeler is asked to read through all the micriblog messages in the topic first. Afterwards, they are given one of the opinion target groups of the topic and three different summaries to score. Note that the labeler does not know which summary is generated by our method and which summaries are generated by the baseline methods. We randomly selected 5 topics in the dataset for the evaluation which contain a total of 41 opinion target groups.

We show the average scores of the results from the three labelers in Table 7. Our method gets the highest scores on all the four aspects and outperforms the two baseline methods by a large margin. For the overall score OR, we get an incearement of two compared to LexRank and 0.82 compared to ILP. One important reason that our method achieves much better results is that we consider the importance of both sentences and opinion targets. For each opinion target group, there exists some wrong opnion targets introucd by the error of extraction or clusting. Our co-ranking method enables us to focus only on the important targets to generate the summary while LexRand and ILP always select sentences from unrelated messages.

7.4 Case Study

Table 8 shows an example of our opinion target group and its summary. We select the most frequently appeared opinion target group of the topic #彭宇承认撞了南京老太 # (#Peng Yu admitted to knocking over the Nanjing Granny#). Two opinion targets which get the highest scores after the co-ranking algorithm are selected as the representative labels for the group. Two microblog sentences are used as opinion summary.

The opinion summary generated by CMiner can provide many insights for the users. It helps users to get a deeper understanding of the targets and people's opinion on the topic. Our CMiner system provides a powerful and elegant way to understand the opinions embedded in thousands of microblog messages.

8 Conclusion and Future Work

In this paper, we present an opinion mining system for Chinese microblogs called CMiner. We propose novel algorithms for opinion target extraction and opinion summarization, both of which have not been well investigated in microblogs yet. We integrate the techniques into an end-to-end system which can help to effectively understand the users' opinion towards different targets in a topic. For opinion target extraction, an unsupervised label propagation algorithm is proposed to collectively rank the opinion target candidates of all sentences in a topic. We also propose

TABLE 8
Case Study of Opinion Target Clustering and Summarization*

Representative Opinion Targets
法官 (judge) 法律 (law)
Opinion Target Group
法官 (judge) 法律 (law) 官方 (official) 司法 (justice) 说法 (statement) 法官的审判(the judgment of the judge)...
Opinion Summary
1. 问题的关键根本就不在于彭宇撞没撞人, 而是法官的判决推理否决掉了一切见义勇为乐于助人的良好道德品质, 照他的判决推理, 谁扶了人就是谁撞的, 谁送人去医院就是谁撞的, 那这个人就应该赔偿, 照他的推理:帮助 = 撞人 = 赔偿!! (Whether Peng Yu hit the granny or not is not the crux of the problem. The judge's verdict vetoes the good moral character. According to his logic, the one who helps the victim is the offender, and this person should pay the compensation, which means: help = hit = compensate!!)
2. 为掩盖自己的错误, 最好的办法是让别人来承担责任□这是中国官场上的一贯作风, 只能证明法律是不可靠的, 要想用法律去维护自己的合法权益必须有深厚的关系, 或者有共同的利益。(The common Chinese way to cover up one's own mistake is to make somebody else take the responsibility. The case can only prove that the law is not reliable. If people want to use the law to protect their legitimate rights and interests, they must be well-connected, or have a common interest with the authorities.)

**The topic selected here is #彭宇承认撞了南京老太 # (#Peng Yu admitted to knocking over the Nanjing Granny#) which aroused a heated discussion on microblog websites. Only one opinion target group and its summary is displayed due to space limit. The background of this topic can be found at https://en.wikipedia.org/wiki/Xu_Shoulan_v._Peng_Yu*

a dynamic programming based algorithm for segmenting Chinese hashtags. For opinion summarization, this is the first study that performs aspect-based opinion summarization on microblog texts. Addressing the diversity issue of opinion targets in a microblog topic, we choose to cluster numerous opinion targets into several groups. A co-ranking algorithm is used to generate both representative targets and summary sentences for each group.

The study presents a preliminary exploration of mining opinion in microblog topics. There is still some weakness in our system which will be addressed in future work. First of all, the opinion target extraction performance is far from perfect. One important reason is that the basic natural language processing techniques such as Chinese word segmentation and part-of-speech tagging cannot satisfy our demands. These natural language processing tasks on microblogs have also attracted a lot of attention recently [41]. We will keep tracking the state-of-the-art techniques and try to incorporate them into our system. Secondly, the label propagation algorithm for opinion target extraction can only find one opinion target for each sentence. An intuitive way for finding multiple targets is to set a threshold for the candidates' scores. But in fact, it does not show good performance in the experiments. We think that the syntactic information of microblog sentences might help to decide the number of opinion targets, but the tradeoff between the benefit and the noise introduced by syntactic analysis is still difficult. In addition, our system ranks microblog sentences to generate opinion summary. Each sentence can be quite long and different sentences from different users are isolated. It will be more concise if we can produce the summary in the sub-sentence or even phrase level and

reorganize them into a coherent paragraph. Lastly, personal influence has been widely studied for online social networks [42], [43]. The messages posted by influential users will be more important and should have higher probabil ity to be included in the summary. Measuring user's influence in a microblog topic and generating influence-sensitive opinion summary can be an interesting topic to research in the future.

Acknowledgments

The authors thank Jiwei Tan, Jin'ge Yao, and Jianmin Zhang for producing the baseline results and labeling the summaries. This work was supported by National Hi-Tech Research and Development Program (863 Program) of China (2015AA015403, 2014AA015102) and National Natural Science Foundation of China (61331011). Xiaojun Wan is the corresponding author.

References

[1] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proc. 23rd Int. Conf. Comput. Linguistics: Posters. Assoc. Comput. Linguistics*, 2010, pp. 36–44.

[2] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics: Human Language Technol.*, 2011, vol. 1, pp. 151–160.

[3] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph," in *Proc. First Workshop Unsupervised Learning NLP Assoc. Comput. Linguistics*, 2011, pp. 53–63.

[4] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures Human Language Technol.*, vol. 5.1, pp. 1–167, 2012.

[5] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach," in *Proc. 20th ACM Int. Conf. Inform. Knowl. Manag.*, pp. 1031–1040, 2011.

[6] L. Zhuang, F. Jing, and X. Zhu, "Movie review mining and summarization," in *Proc. ACM 15th Conf. Inform. Knowl. Manag.*, Arlington, Virginia, USA, Nov. 2006, pp. 43–50.

[7] N. Jakob and I. Gurevych, "Extracting opinion targets in a single- and cross-domain setting with conditional random fields," in *Proc. Conf. Empirical Methods Natural Language Process., Assoc. Comput. Linguistics*, 2010, pp. 1035–1045.

[8] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[9] F. Li, S.J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proc. 50th Annu. Meet. Assoc. Comput. Linguistics*, Jeju, Republic of Korea, Jul. 8–14 2012, pp. 410–419.

[10] X. Liu K. Li, M. Zhou, and Z. Xiong, "Collective semantic role labeling for tweets with clustering," in *Proc. 22nd Int. Joint Conf. Artificial Intell., Volume Volume Three*, 2011, pp. 1832–1837.

[11] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining ACM*, 2004, pp. 168–177.

[12] S. M. Kim and E. Hovy, "Extracting opinions, opinion holders and topics expressed in online news media text," in *Proc. ACL Workshop Sentiment Subjectivity Text*, 2006, pp. 1–8.

[13] G. Dray, M. Plantié, A. Harb, P. Poncelet, M. Roche and F. Trousset, "Opinion mining from blogs," *Int. J. Comput. Inform. Syst. Ind. Manag. Appl.*, vol. 1, pp. 205–213, 2009.

[14] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," *Urbana*, vol. 51, p. 61801, 2008.

[15] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2010, pp. 56–65.

[16] T. Ma and X. Wan, "Opinion target extraction in Chinese news comments," in *Proc. 23rd Int. Conf. Comput. Linguistics: Posters. Assoc. Comput. Linguistics*, 2010, pp. 782–790.

[17] B. Yang and C. Claire, "Joint inference for fine-grained opinion extraction," in *Proc. 51st Annu. Meet. Assoc. Comput. Linguistics*, pp. 1640–1649, 2013.

[18] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources Eval.*, vol. 39, no. 2-3, pp. 165–210, 2005.

[19] H.D. Kim, K. Ganesan, P. Sondhi, and C. Zhai, "Comprehensive review of opinion summarization," Tech. Rep., University of Illinois at Urbana-Champaign, 2011.

[20] L. Bing, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the web," in *Proc. 14th Int. Conf. World Wide Web ACM*, 2005, pp. 342–351.

[21] G. Carenini, J. C. K. Cheung, and A. Pauls, "Multi-document summarization of evaluative text," *Comput. Intell.*, vol. 29.4, pp. 545–576, 2013.

[22] H. D. Kim, and C. Zhai, "Generating comparative summaries of contradictory opinions in text," in *Proc. 18th ACM Conf. Inform. Knowl. Manag.*, 2009, pp. 385–394.

[23] Y. Lu, H. Duan, H. Wang, and C. Zhai, "Exploiting structured ontology to organize scattered online opinions," in *Proc. 23rd Int. Conf. Comput. Linguistics* , 2010, pp. 734–742.

[24] J. Weng, C. Yang, B. Chen, Y. Wang, and S. Lin, "IMASS: An intelligent microblog analysis and summarization system," *ACL (Syst. Demonstrations)*, 2011, pp. 133–138

[25] N. N. Bora, "Summarizing public opinions in tweets. Journal" *Int. J. Comput. Linguistics Appl.*, vol. 3, no. 1, pp. 41–55, 2012.

[26] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, "Entity-centric topic-oriented opinion summarization in twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 379–387.

[27] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. Int. Conf. Web Search Data Mining ACM*, 2008, pp. 231–240.

[28] J. F. Silva and G. P. Lopes, "A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora," in *Proc. 6th Meet. Math. Language*, 1999, pp. 369–381.

[29] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B. Lee, "Twiner: Named entity recognition in targeted twitter stream," in *Proc. 35th Int. ACM SIGIR Conf. Res. Development Inform. Retrieval ACM*, 2012, pp. 721–730.

[30] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *CMU CALD*, Pittsburgh, PA, *Tech. Rep.*, 2002.

[31] P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 442–457.

[32] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[33] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, vol. 7, 2007, pp. 1606–1611.

[34] B.J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, 2007, pp. 972–976.

[35] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles, "Co-ranking authors and documents in a heterogeneous network", in *Proc. 7th IEEE Int. Conf. Data Mining*, 2007, pp. 739–744.

[36] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479. 2004.

[37] R. McDonald, "A study of global inference algorithms in multi-document summarization," in *Proc. 29th Eur. Conf. IR Res.*, 2007, pp. 557–564.

[38] H. T. Dang, "Overview of duc 2006," presented at the *Document Undestanding Conf.*, Rouchester, NY, USA, 2006.

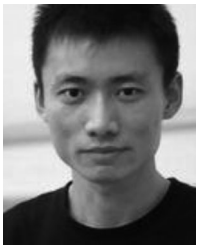
[39] R. Johansson and A. Moschitti, "Syntactic and semantic structure for opinion expression detection," in *Proc. 14th Conf. Comput. Natural Language Learn. Assoc. Comput. Linguistics*, 2010, pp. 67–76.

[40] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meet. Assoc. Comput. Linguistics*, 2006, pp. 433–440.

[41] H. Duan, Z. Sui, Y. Tian, and W. Li, "The cips-sighan clp 2012 Chinese word segmentation on microblog corpora bakeoff," in *Proc. 2nd CIPS-SIGHAN Joint Conf. Chinese Language Process.*, 2012, pp. 35–40.

[42] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy," *ICWSM*, vol. 10, 2010, pp. 10–17.
[43] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in *Proc. 4th ACM Int. Conf. Web Search Data Mining ACM*, 2011, pp. 65–74.

Xinjie Zhou received the BS degree from Beijing University of Posts and Telecommunications in 2012. He is currently working toward the PhD degree at Institute of Computer Science and Technology of Peking University, Beijing, China. His research interests include sentiment analysis and natural language processing.



Xiaojun Wan received the BS, MS and PhD degrees from Peking University in 2000, 2003 and 2006, respectively. He is currently a professor at the Institute of Computer Science and Technology of Peking University, Beijing, China. His research interests include natural language processing and text mining. He has published more than 60 publications in major international conferences and journals, including ACL, SIGIR, AAAI, IJCAI, COLING, EMNLP, ICDM, CIKM, *ACM TOIS*, *Computational Linguistics*, *JASIST*, *KAIS*, *Information Sciences*, and so on. He has served as a PC member or an area chair of major conferences such as ACL, SIGIR, EMNLP, COLING, CIKM, IJCNLP, and NLPCC.

Jianguo Xiao received the MS degree from the Department of Computer Science and Technology of Peking University, Beijing, China, in 1990. He is a professor at the Institute of Computer Science and Technology of Peking University, Beijing, China, and he is also the director of the institute. His research interests include natural language processing and Chinese computing.